

AMENDMENTS TO THE CLAIMS

With respect to the objections and rejections to the claims, the applicant hereby amends the claims as shown hereinafter. The applicant respectfully asserts that there has been no new matter added due to these amendments. Accordingly, please amend the claims as follows:

1. (Currently amended) A method for monitoring application exceptions events generated by a software application, comprising:

operating the software application to generate exception events data responsive to an application exception events;

dynamically identifying said application exception generated by the software application prior to said application exception being logged ~~monitoring the software application to identify an occurrence of said exception event and to obtain said exception event data;~~

obtaining exception data responsive to said application exception;

examining said exception event data, prior to said application exception being logged, to determine whether said application exception event is a critical exception event and to identify critical exception event data;

determining type of said critical exception event; and

processing said critical exception event data responsive to said type of said critical exception event.

2. (Original) The method of Claim 1, wherein said operating includes operating said software application in at least one of a .NET framework and a J2EE framework.

3. (Currently amended) The method of Claim 1, wherein said determining includes determining whether said type of said critical exception event is at least one of a primary critical exception event and a derived critical exception event.

4. (Currently amended) The method of Claim 3, wherein said processing includes processing said critical exception **event** data responsive to at least one of said primary critical exception **event** and said derived critical exception **event**.

5. Cancelled

6. Cancelled

7. (Currently amended) The method of Claim 1, wherein said examining includes examining said critical exception **event** data to determine if an exception chain exists.

8. (Currently amended) The method of Claim 7, wherein said processing further includes collecting critical exception **event** data responsive to said critical exception **event** and creating an exception **event** information database.

9. (Currently amended) The method of Claim 8, wherein said processing further includes creating said critical exception chain and associating said collected critical exception **event** data with said critical exception chain.

10. (Currently amended) The method of Claim 7, wherein said processing includes associating said collected critical exception **event** data with said critical exception chain.

11. (Currently amended) The method of Claim 1, wherein said examining further includes comparing said exception **event** data with data contained within an exception **event** information database to determine whether said exception **event** is said critical exception **event**.

12. (Currently amended) The method of Claim 1, wherein said examining further includes labeling said exception **event** as at least one of a critical exception, a non-critical exception, a derived exception **event** and a primary exception **event**.

13. (Currently amended) The method of Claim 12, wherein said processing further includes updating said exception ~~event~~ information database with said exception ~~event~~ data

14. (Currently amended) A system for operating a software application in a predefined framework, wherein the system includes a processor configured to execute machine-readable computer program code including instructions for causing a controller to implement a method for monitoring an application exception ~~events~~ generated by the software application, the method comprising, comprising:

operating the software application to generate exception ~~event~~ data responsive to an application exception ~~event~~;

dynamically identifying said application exception generated by the software application prior to said application exception being logged ~~monitoring the software application to identify an occurrence of said exception event and to obtain said exception event data~~;

obtaining exception data responsive to said application exception;

examining said exception ~~event~~ data, prior to said application exception being logged, to determine whether said application exception ~~event~~ is a critical exception ~~event~~ and to identify critical exception ~~event~~ data;

determining type of said critical exception ~~event~~; and

processing said critical exception ~~event~~ data responsive to said type of said critical exception ~~event~~.

15. (Currently amended) The system of Claim 14, wherein said examining includes examining said critical exception ~~event~~ data to determine whether said critical exception ~~event~~ is at least one of a primary critical exception ~~event~~ and a derived critical exception ~~event~~.

16. (Original) The system of Claim 14, wherein said predefined framework is at least one of a .NET framework and a J2EE framework.

17. Cancelled

18. (Currently amended) The system of Claim 14, wherein said processing includes processing said critical exception ~~event~~ data responsive to at least one of said primary critical exception ~~event~~ and said derivative critical exception.

19. (Currently amended) A computer-readable storage medium encoded with a machine-readable computer program code, the program code including instructions for causing a controller to implement a method for monitoring application exceptions ~~events~~ generated by a software application, the method comprising:

operating the software application to generate exception ~~event~~ data responsive to an application exception ~~event~~;

~~dynamically identifying said application exception generated by the software application prior to said application exception being logged monitoring the software application to identify an occurrence of said exception event and to obtain said exception event data;~~

obtaining exception data responsive to said application exception;

examining said exception event data, prior to said application exception being logged, to determine whether said application exception ~~event~~ is a critical exception ~~event~~ and to identify critical exception ~~event~~ data;

determining type of said critical exception ~~event~~; and

processing said critical exception ~~event~~ data responsive to said type of said critical exception ~~event~~.

20. Cancelled